

# Top down e bottom up

**Top down** e **bottom up** sono termini di lingua inglese attribuiti a **metodologie** adoperate per analizzare **situazioni problematiche** e costruire **ipotesi** adeguate alla loro **soluzione**: il concetto di **situazione problematica** è riconducibile ad ambiti tra i più vari come ad esempio la elaborazione di un **programma informatico**, la risoluzione di un problema geometrico ovvero matematico, la elaborazione di un testo, la risoluzione di un problema pratico/operativo.

## Top down



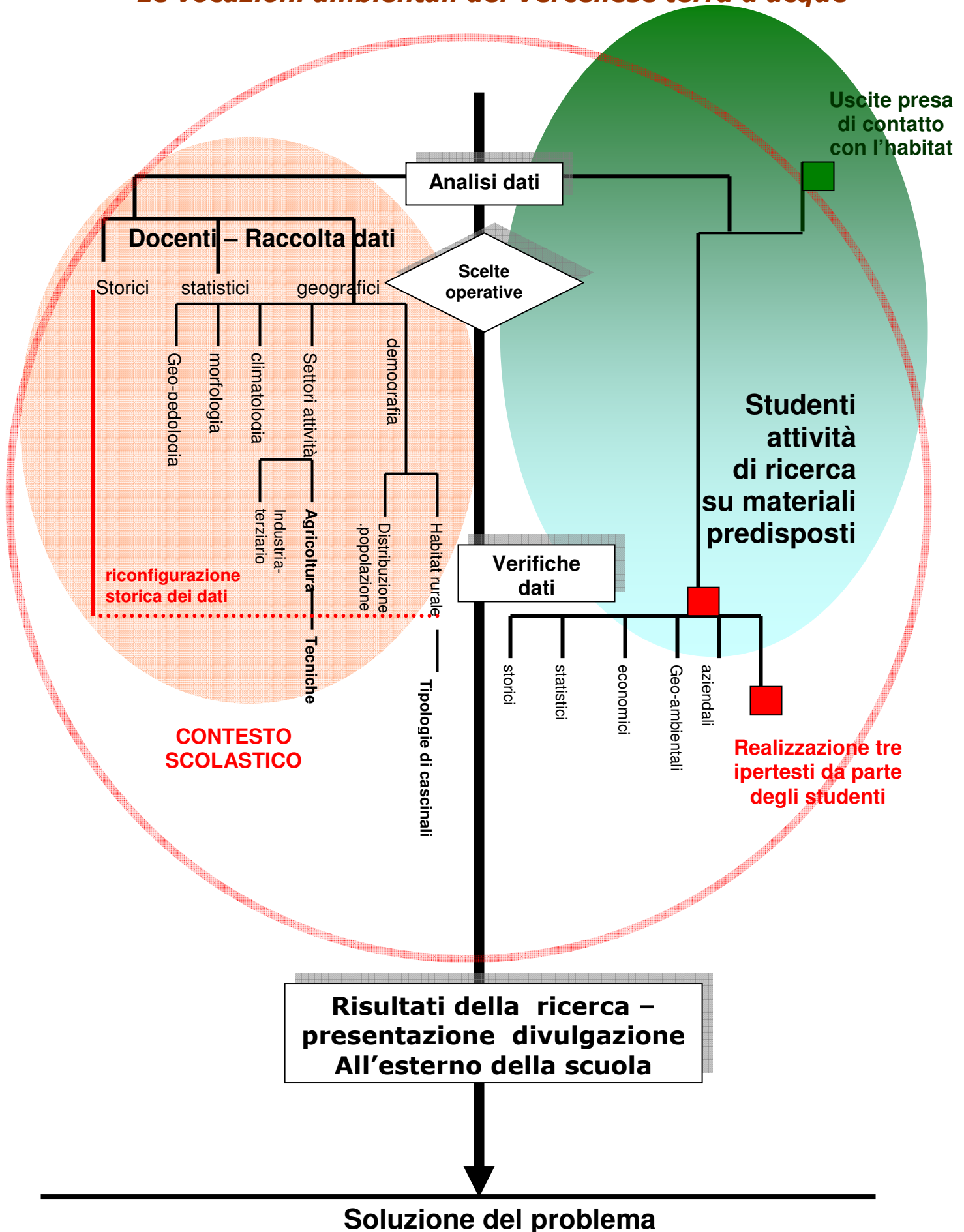
Il *top down* richiama l'immagine di una **piramide** con la **cima (top)** in alto e la **base (down)** posta in posizione **orizzontale**; la caratterizzazione dinamica del metodo prevede di partire dal *top* e scendere verso il *down*. **Top**, termine molto usato nel linguaggio contemporaneo ( *top ten, top model, ecc.*), può essere efficacemente tradotto in **obiettivo** inteso come obiettivo da raggiungere per risolvere una situazione problematica; **down** è riconducibile alla **strategia** necessaria a conseguire il fine predeterminato.

Il top down parte dall'obiettivo e da esso fa scaturire la strategia direttamente adatta a determinare l'obiettivo stesso, quindi valorizza il perché e da esso fa dipendere il come, ovvero la strategia; individua, quindi, le risorse necessarie, precisa quelle disponibili e identifica quelle mancanti, propone successivamente ogni risorsa mancante come sub-obiettivo ovvero come sotto-problema in cui ciascun sub-obiettivo richiede una sub-strategia ad esso correlata.

Il *top down* struttura e sistematizza la riflessione, motiva ogni passaggio in modo logico e ripulisce il metodo di lavoro da tentativi casuali, intreccia strettamente analisi ed elaborazione, dà alla strategia complessiva un carattere di eleganza sostanziale e formale che si estrinseca come trasparenza e sistematicità. È soprattutto adatto, ma non solo, a situazioni complesse, inedite, di differenti tipologie di problemi.

# Un esempio di top down: come studiare un problema di storia locale

## *Le vocazioni ambientali del Vercellese terra d'acque*

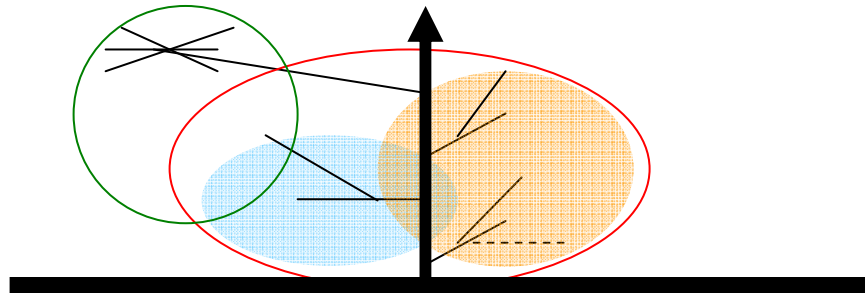


## Bottom up



Il *bottom up* richiama invece un'immagine raffigurante una freccia in cui la coda è il bottom (la parte bassa) mentre up è la punta: dal punto di vista dinamico si parte dal *bottom* e si procede verso *up*.

Il **bottom up** prende corpo dal punto di partenza (*bottom*) ovvero dalla situazione iniziale; considera l'obiettivo finale, induce a costruire un **percorso sequenziale** organizzato in passaggi successivi in cui l'ancoraggio tra traguardi intermedi e obiettivo finale è ricercato con metodo generalmente improntato a **tentativo ed errore** quindi di tipo **casuale**, nei casi migliori **intuitivo**.



### La testualità come rappresentazione del reale

È per lo più adatto a **situazioni problematiche lineari, semplici**, delle quali la **dinamica è generalmente nota**.

**Top-down** e **bottom-up** sono **strategie di elaborazione dell'informazione** e di gestione delle conoscenze, riguardanti principalmente la scelta di particolari software e, per estensione, fanno riferimento ad altre teorie umanistiche e a teorie dei sistemi.

Nel modello **top-down** è formulata una **visione generale del sistema** senza scendere nel dettaglio di alcuna delle sue parti. **Ogni parte del sistema è successivamente rifinita** aggiungendo maggiori dettagli dalla progettazione. Ogni nuova parte così ottenuta **può quindi essere nuovamente rifinita**, specificando ulteriori dettagli finché la *specifica completa* è sufficientemente dettagliata da validare il modello. Il modello top-down è spesso progettato con l'ausilio di scatole nere che semplificano il riempimento ma non consentono di capirne il meccanismo elementare.

In contrasto con il modello top-down c'è la progettazione **bottom-up**, nella quale parti individuali del sistema sono **specificate in dettaglio**. Queste parti vengono poi **connesse** tra

loro in modo da formare componenti più grandi, che vengono a loro volta **interconnessi** fino a realizzare un sistema completo. Le strategie basate sul flusso informativo bottom-up sembrano potenzialmente necessarie e sufficienti, dato che sono basate sulla **conoscenza** di tutte le **variabili** che possono condizionare gli elementi del sistema.

---

In sintesi

Top-down: Un generico problema viene scomposto in tanti sottoproblemi di più semplice risoluzione, e ad ognuno è associato un sottoprogramma.

## Informatica

Nel processo di sviluppo software, gli approcci **top-down** e **bottom-up** giocano un ruolo fondamentale. L'approccio top-down enfatizza la pianificazione ed una completa comprensione del sistema. È ovvio che nessuna codifica può iniziare finché non si è raggiunto almeno un sufficiente livello di dettaglio nella progettazione di una parte significativa del sistema. Questo, comunque, ritarda la fase di test delle ultime unità funzionali di un sistema finché una parte significativa della progettazione non è stata completata. L'approccio bottom-up enfatizza la codifica e la fase di test precoce, che può iniziare appena il primo modulo è stato specificato. Questo approccio, comunque, induce il rischio che i moduli possano essere codificati senza avere una chiara idea di come dovranno essere connessi ad altre parti del sistema, e quel tipo di link potrebbe non essere facile. La riusabilità del codice è uno dei principali benefici dell'approccio bottom-up. La progettazione top-down è stata sostenuta negli anni 70 dai ricercatori IBM Harlan Mills e Niklaus Wirth. Mills sviluppò i concetti della programmazione strutturata per uso pratico e li testò in un progetto del 1969 per automatizzare l'archivio del *New York Times*. Il successo ingegneristico e gestionale di questo progetto condusse alla crescita dell'approccio top-down tramite IBM ed il resto dell'industria informatica. Niklaus Wirth, che tra altre imprese sviluppò il linguaggio di programmazione Pascal, scrisse l'autorevole documento *Lo sviluppo del software per raffinamenti successivi*. I metodi top-down erano i preferiti nell'ingegneria del software negli anni 80. I moderni approcci alla progettazione software tipicamente combinano sia la tecnica top-down che quella bottom-up. Sebbene la comprensione del sistema completo è tipicamente considerata necessaria per una buona progettazione che conduce teoricamente ad un approccio top-down, la maggior parte dei progetti software cercano di fare uso di codice già esistente ad alcuni livelli. I moduli pre-esistenti danno alla progettazione una tendenza bottom-up. Alcuni approcci di progettazione operano progettando un sistema parzialmente funzionale che viene completamente codificato, poi questo sistema viene quindi espanso fino a soddisfare tutti i requisiti di progetto.

## Programmazione

---

La programmazione top-down è uno stile di programmazione, fondamento dei tradizionali linguaggi procedurali, nel quale la progettazione inizia specificando parti complesse e suddividendole successivamente in parti più piccole. Eventualmente, i componenti sono specificati quanto basta per la codifica ed il programma viene anche scritto. Questo è l'esatto opposto della programmazione bottom-up che è comune nei linguaggi orientati agli oggetti come C++ o Java. La tecnica per la scrittura di un programma mediante l'utilizzo dei metodi top-down indica di scrivere una procedura principale che indica dei nomi per le principali funzioni di cui avrà bisogno. In seguito, il gruppo di programmazione esaminerà i requisiti di ognuna di queste funzioni ed il processo verrà ripetuto. Queste sotto-procedure a comparto eseguiranno eventualmente azioni così semplici che porteranno ad una codifica semplice e

concosa. Quando tutte le varie sotto-procedure sono state codificate, il programma è realizzato.

## **Vantaggi**

### **Il gruppo di programmazione resta focalizzato sull'obiettivo.**

- Ognuno conosce il proprio compito.
- Nel momento in cui parte la programmazione, non vi sono più domande.
- Il codice è semplice da seguire, dato che è scritto in maniera metodica e con uno scopo preciso.

## **Svantaggi**

- La programmazione top-down può complicare la fase di test, dato che non esisterà un eseguibile finché non si arriverà quasi alla fine del progetto.
- La programmazione bottom-up agevola il test di unità, ma finché il sistema non si unisce non può essere testato nella sua interezza, e ciò causa spesso complicazioni verso la fine del progetto "Individualmente ci siamo, insieme falliamo."
- Tutte le decisioni dipendono dall'avvio del progetto ed alcune decisioni non possono essere fatte sulla base del dettaglio delle specifiche.

## **Neuroscienza e psicologia**

Questo lessico è anche utilizzato nella neuroscienza e nella psicologia. Lo studio dell'attenzione visiva ne è un esempio. Se la tua attenzione è rivolta ad un fiore in un campo, può semplicemente essere che il fiore è visivamente più rilevante rispetto al resto del campo. L'informazione che ti ha portato ad osservare il fiore ti è giunta in modo bottom-up. La tua attenzione non è stata condizionata dalla conoscenza del fiore; gli stimoli esterni era già propriamente sufficienti. Confronta questa situazione con una in cui tu stai cercando un fiore. Hai una rappresentazione di cosa cerchi. Quando vedi l'oggetto che cerchi, questo è saliente. Questo è un esempio dell'uso dell'informazione in modo top-down.

Parte delle informazioni sono tratte da *Wikipedia*